Published since January 1985 and Published before September 2003
Terms used
store restore instructions register determine execution order before prior

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanc
Try this search

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 ne
Best 200 shown R

1 The priority-based coloring approach to register allocation
Fred C. Chow, John L. Hennessy
October 1990 **ACM Transactions on Programming Languages and Sys**
Volume 12 Issue 4
Publisher: ACM Press
Full text available: pdf(2.97 MB) Additional Information: full citation, abst
citings, index ter

Global register allocation plays a major role in determining the efficacy compiler. Graph coloring has been used as the central paradigm for regis modern compilers. A straightforward coloring approach can suffer from shortcomings. These shortcomings are addressed in this paper by colorin priority ordering. A natural method for dealing with the spilling emerges The detailed algorithms for a priority-based colori ...

2 Parallel execution of prolog programs: a survey
Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V.
July 2001 **ACM Transactions on Programming Languages and System**
Volume 23 Issue 4
Publisher: ACM Press
Full text available: pdf(1.95 Additional Information: full citation, abst

MB) citings, index ter

Since the early days of logic programming, researchers in the field realiz exploitation of parallelism present in the execution of logic programs. Tl nature, the presence of nondeterminism, and their referential transparenc characteristics, make logic programs interesting candidates for obtaining parallel execution. At the same time, the fact that the typical applications programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic pr parallelism, prolog

3 Compiler transformations for high-performance computing

David F. Bacon, Susan L. Graham, Oliver J. Sharp

December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Publisher: ACM Press

Full text available: pdf(6.32 MB) Additional Information: full citation, abst citings, index ter

In the last three decades a large number of compiler transformations for programs have been implemented. Most optimizations for uniprocessors of instructions executed by the program using transformations based on scalar quantities and data-flow techniques. In contrast, optimizations for superscalar, vector, and parallel processors maximize parallelism and me transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, parallelism, superscalar processors, vectorization

4 An abstract machine for tabled execution of fixed-order stratified logic pro

Konstantinos Sagonas, Terrance Swift

May 1998 **ACM Transactions on Programming Languages and System** Volume 20 Issue 3

Publisher: ACM Press

Full text available: pdf(602.38 KB) Additional Information: full citation, abst citings, index ter

SLG resolution uses tabling to evaluate nonfloundering normal logic pr (the well-founded semantics. The SLG-WAM, which forms the engine of can compute in-memory recursive queries an order of magnitute faster th deductive databases. At the same time, the SLG-WAM tightly intergrate tabled SLG code, and executes Prolog code with minimal overhead com As a result, the SLG-WAM brings to logic progr ...

Keywords: SLG, WAM, memoing, prolog, stratification theories, tablin

5 Register integration: a simple and efficient implementation of squash reuse

Amir Roth, Gurindar S. Sohi

December 2000 **Proceedings of the 33rd annual ACM/IEEE internatio Microarchitecture**

Publisher: ACM Press

Full text available: pdf(154.98 KB) ps (573.81 KB) Publisher Site

Additional Information: full citation, refe index terms

6 Sentinel scheduling: a model for compiler-controlled speculative execution

Scott A. Mahlke, William Y. Chen, Roger A. Bringmann, Richard E. Hank Hwu, B. Ramakrishna Rau, Michael S. Schlansker

November 1993 **ACM Transactions on Computer Systems (TOCS)**, Vc

Publisher: ACM Press

Full text available: pdf(2.26 MB)

Additional Information: full citation, abst citings, index ter

Speculative execution is an important source of parallelism for VLIW ar processors. A serious challenge with compiler-controlled speculative exe efficiently handle exceptions for speculative instructions. In this article, features and compile-time scheduling support collectively referred to as is introduced. Sentinel scheduling provides an effective framework for b controlled speculative executi ...

Keywords: VlIW processor, exception detection, exception recovery, in

scheduling, instruction-level parallelism, speculative execution, supersca

7 Register renaming and dynamic speculation: an alternative approach
Mayan Moudgill, Keshav Pingali, Stamatis Vassiliadis
December 1993 **Proceedings of the 26th annual international symposiu Microarchitecture**
Publisher: IEEE Computer Society Press
Full text available: pdf(1.49 MB) Additional Information: full citation, refe

8 Implementation of precise interrupts in pipelined processors
James E. Smith, Andrew R. Pleszkun
June 1985 **ACM SIGARCH Computer Architecture News , Proceeding annual international symposium on Computer architecture** 13 Issue 3
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(893.17 KB) Additional Information: full citation, citir

9 Experience with a software-defined machine architecture
David W. Wall
May 1992 **ACM Transactions on Programming Languages and System** Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(2.86 MB) Additional Information: full citation, abst citings, index ter

We have built a system in which the compiler back end and the linker w present an abstract machine at a considerably higher level than the actua intermediate language translated by the back end is the target language o compilers and is also the only assembly language generally available. Th intermodule register allocation, which would be harder if some of the co had come from a traditional assembler, out of sight of ...

Keywords: RISC, graph coloring, intermediate language, interprocedura pipeline scheduling, profiling, register allocation, register windows

10 The MC88110 implementation of precise exceptions in a superscalar archi
Nasr Ullah, Matt Holle
March 1993 **ACM SIGARCH Computer Architecture News**, Volume 2
Publisher: ACM Press
Full text available: pdf(1.06 MB) Additional Information: full citation, abst terms

This paper describes the precise exception model of the MC88110 symm microprocessor. The MC88110 is a superscalar, pipelined processor that exection units and allows out-of order execution of instructions. The MC fully precise exceptions and presents the architecturally correct state to a handling routine in a manner that minimizes exception response latency. latency timings in the MC88110 are described, and several ...

11 Exploiting instruction level parallelism in processors by caching scheduled
Ravi Nair, Martin E. Hopkins
May 1997 **ACM SIGARCH Computer Architecture News , Proceeding annual international symposium on Computer architecture** 25 Issue 2
Publisher: ACM Press
Full text available: pdf(2.01 MB) Additional Information: full citation, abst citings, index ter

Modern processors employ a large amount of hardware to dynamically d single-threaded programs and maintain the sequential semantics implied The complexity of some of this hardware diminishes the gains due to pa longer clock period or increased pipeline latency of the machine.In this p processor implementation which dynamically schedules groups of instru executing them on a fast simple engine and caches them f ...

12 Compiler techniques for code compaction
Saumya K. Debray, William Evans, Robert Muth, Bjorn De Sutter
March 2000 **ACM Transactions on Programming Languages and Syste** Volume 22 Issue 2

Publisher: ACM Press
Full text available: pdf(409.20 KB) Additional Information: full citation, abst citings, index ter

In recent years there has been an increasing trend toward the incorpor ati into a variety of devices where the amount of memory available is limite desirable to try to reduce the size of applications where possible. This ar use of compiler techniques to accomplish code compaction to yield smal main contribution of this article is to show that careful, aggressive, inter optimization, together with procedural abstr ...

Keywords: code compaction, code compression, code size reduction

13 VLIW compilation techniques in a superscalar environment
Kemal Ebcioglu, Randy D. Groves, Ki-Chang Kim, Gabriel M. Silberman,
June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD** Issue 6
Publisher: ACM Press
Full text available: pdf(1.30 MB) Additional Information: full citation, abst citings, index ter

We describe techniques for converting the intermediate code representat program, as generated by a modern compiler, to another representation v same run-time results, but can run faster on a superscalar machine. The a novel parallelization techniques for Very Long Instruction Word (VLIW and place together independently executable operations that may be far a code. i.e., they may be se ...

Keywords: VLIW, compiler optimizations, global scheduling, profiling software pipelining, superscalars

14 A novel renaming scheme to exploit value temporal locality through physi and unification
Stephen Jourdan, Ronny Ronen, Michael Bekerman, Bishara Shomar, Adi
November 1998 **Proceedings of the 31st annual ACM/IEEE internation Microarchitecture**

Publisher: IEEE Computer Society Press
Full text available: pdf(2.41 MB) Additional Information: full citation, refe index terms

Keywords: dependency redirection, physical register reuse, register and result reuse, value temporal locality

15 Optimization for a superscalar out-of-order machine
Anne M. Holler
December 1996 **Proceedings of the 29th annual ACM/IEEE internation Microarchitecture**
Publisher: IEEE Computer Society
Full text available: pdf(1.55 MB) Additional Information: full citation, abst terms

Compiler optimization plays a key role in unlocking the performance of innovative dynamically-scheduled machine which is the first implement PA 2.0 member of the HP PA-RISC architecture family. This wide supe order machine provides significant execution bandwidth and automatical runtime; however, despite its ample hardware resources, many of the opt transformations which proved effective for the PA-8000 served to au ...

16 Instruction-level reverse execution for debugging
Tankut Akgul, Vincent J. Mooney
November 2002 **ACM SIGSOFT Software Engineering Notes , Proceed ACM SIGPLAN-SIGSOFT workshop on Program ana tools and engineering PASTE '02**, Volume 28 Issue 1
Publisher: ACM Press
Full text available: pdf(241.83 KB) Additional Information: full citation, abst citings, index ter

The ability to execute a program in reverse is advantageous for shortenir paper presents a reverse execution methodology at the assembly instruct memory and time overheads. The core idea of this approach is to genera able to undo, in almost all cases, normal forward execution of an assemb program being debugged. The methodology has been implemented on a

in a custom made debugger. Compared to previous w ...

Keywords: debugging, reverse code generation, reverse execution

17 Software pipelining

Vicki H. Allan, Reese B. Jones, Randall M. Lee, Stephen J. Allan
September 1995 **ACM Computing Surveys (CSUR)**, Volume 27 Issue 3
Publisher: ACM Press
Full text available: pdf(4.72 MB) Additional Information: full citation, abst citings, index ter

Utilizing parallelism at the instruction level is an important way to impr Because the time spent in loop execution dominates total execution time optimizations focuses on decreasing the time to execute each iteration. S is a technique that reforms the loop so that a faster execution rate is reali executed in overlapped fashion to increase parallelism.Let {ABC}n
Keywords: instruction level parallelism, loop reconstruction, optimizati pipelining

18 Multithreading and value prediction: Speculative lock elision: enabling hig multithreaded execution

Ravi Rajwar, James R. Goodman
December 2001 **Proceedings of the 34th annual ACM/IEEE internation Microarchitecture**
Publisher: IEEE Computer Society
Full text available: pdf(1.37 MB) Publisher Site Additional Information: full citation, abst citings

Serialization of threads due to critical sections is a fundamental bottlene performance in multithreaded programs. Dynamically, such serialization unnecessary because these critical sections could have safely executed c locks. Current processors cannot fully exploit such parallelism because t mechanisms to dynamically detect such false inter-thread dependences.V *Speculative Lock Elision (SLE),* a novel micro-architectura ...

19 Avoidance and suppression of compensation code in a trace scheduling cor

Stefan M. Freudenberger, Thomas R. Gross, P. Geoffrey Lowney

July 1994 **ACM Transactions on Programming Languages and System** Volume 16 Issue 4

Publisher: ACM Press

Full text available: pdf(3.58 MB) Additional Information: full citation, abst citings, index ter

Trace scheduling is an optimization technique that selects a sequence of trace and schedules the operations from the trace together. If an operatio basic block boundaries, one or more compensation copies may be requir code. This article discusses the generation of compensation code in a tra compiler and presents techniques for limiting the amount of compensatic (restricting code motion so that no compensatio ...

Keywords: SPEC89, instruction-level parallelism, performance evaluati

20 Dynamic dead-instruction detection and elimination

J. Adam Butts, Guri Sohi

October 2002 **ACM SIGOPS Operating Systems Review , ACM SIGPL SIGARCH Computer Architecture News , Proceedings o international conference on Architectural support for pr languages and operating systems ASPLOS-X,** Volume 3(10 , 5

Publisher: ACM Press

Full text available: pdf(1.50 MB) Additional Information: full citation, abst citings

We observe a non-negligible fraction--3 to 16% in our benchmarks--of *a instructions,* dynamic instruction instances that generate unused results. these instructions arise from static instructions that also produce useful r compiler optimization (specifically instruction scheduling) creates a sign these *partially dead* static instructions. We show that most of the dynam arise from a small set of st ...

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results **BROWSE SEARCH IEEE GUID**

Results for "**(((determine instruction order instrumenting store restore <in>metadata)) <and> (pyr &g...**"
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History
New Search

Modify Search

(((determine instruction order instrumenting store re

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

» Key

- **IEEE JNL** IEEE Journal or Magazine
- **IEE JNL** IEE Journal or Magazine
- **IEEE CNF** IEEE Conference Proceeding
- **IEE CNF** IEE Conference Proceeding
- **IEEE STD** IEEE Standard

Indexed by Inspec

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results **BROWSE SEARCH IEEE GUID**

Results for "(((instruction execution instrumenting store restore)<in>n <and> (pyr >= 19...

Your search matched **0** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((instruction execution instrumenting store restore):

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

» Key

- **IEEE JNL** IEEE Journal or Magazine
- **IEE JNL** IEE Journal or Magazine
- **IEEE CNF** IEEE Conference Proceeding
- **IEE CNF** IEE Conference Proceeding
- **IEEE STD** IEEE Standard

Indexed by Inspec®

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((scheduling priority instrumenting store restore)<in>me
<and> (pyr >= 1980..."
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc
Descending** order.

» Search Options

View Session History
New Search

Modify Search

(((scheduling priority instrumenting store restore)<ir

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec

Google Scholar BETA

instrumentation instruction scheduling store re | 1980 | - | 2

Scholar **All articles** Recent articles Results **1 - 10** of about **1,620** for **instrur**

All Results
D Scales
K Gharachorloo
P Lowney
J Larus
D Culler

Shasta: a low overhead, software-only approach for supporting fine-grain shared memory - group of 9 »
DJ Scales, K Gharachorloo, CA Thekkath - Proceedings of the seventh international conference on ..., 1996 - portal.acm.org
... be moved upward from the **instrumentation** point within ... the load when the add **instruction** is executed ... compiler is typically successful in **scheduling** the original ...
Cited by 262 - Related Articles - Web Search - BL Direct

Speculative **register** promotion using advanced load address table (ALAT) - group of 5 »
J Lin, T Chen, WC Hsu, PC Yew - Code Generation and Optimization, 2003. CGO 2003. ..., 2003 - ieeexplore.ieee.org
... on the top of ORC to **instrument** the code ... flags are transformed into the corresponding assembly **instructions**. ... code for speculation during list **scheduling** as well ...
Cited by 15 - Related Articles - Web Search

Method of reducing the number of overhead instructions by modifying the program to locate ... - group of 3 »
DJ Scales - US Patent 5,758,183, 1998 - Google Patents
... virtual addresses assigned to the memories are allocated to **store** a shared data structure as one or more blocks accessible by **instructions** ofprograms executing ...
Cited by 33 - Related Articles - Web Search

Instruction scheduling for **instruction** level parallel processors - group of 3 »
P Faraboschi, JA Fisher, C Young - Proceedings of the IEEE, 2001 - ieeexplore.ieee.org
... it is the infamous "job shop **scheduling**" problem [5 ... profiles used to drive **instruction scheduling** as static ... oldest technique is **instrumentation**, where extra ...
Cited by 19 - Related Articles - Web Search - BL Direct

Exploiting hardware performance counters with flow and context sensitive profiling - group of 11 »
G Ammons, T Ball, JR Larus - ACM SIGPLAN Notices, 1997 - portal.acm.org
... have complex microarchitectures that dy- namically **schedule instructions**. ... predictable metrics, such as **instruction** fre- quency ... the effect of **instrumentation** code ...
Cited by 151 - Related Articles - Web Search - BL Direct

The transmeta code morphing software: Using speculation, recovery, and adaptive retranslation to ... - group of 15 »
JC Dehnert, BK Grant, JP Banning, R Johnson, T ... - International Symposium on Code Generation and Optimization, 2003 - doi.ieeecomputersociety.org
... There are **scheduling** constraints that must be ... emulation systems as interpreters (**instruction**-at-a-time ... created for purposes of optimization or **instrumentation**. ...
Cited by 71 - Related Articles - Web Search

Validation checking of shared memory accesses - group of 2 »
DJ Scales - US Patent 5,761,729, 1998 - Google Patents
... of the programs allocates a portion ofthevirtual shared addresses to **store** a shareddata ...

miss handling code is executed before the load **instructions** are executed ...
Cited by 35 - Related Articles - Web Search

Register File Design Considerations in Dynamically Scheduled Processors - group of 10 »
KI Farkas, NP Jouppi, P Chow - Proceedings of the Second IEEE Symposium on High-Performance ..., 1996 - doi.ieeecomputersociety.org
... The **instruction scheduling** logic includes a single dis- patch ... **Instruction** counts are in millions; the "rates ... ing an object code **instrumentation** system called ...
Cited by 75 - Related Articles - Web Search

Fine-grain parallelism with minimal hardware support: a compiler-controlled threaded abstract ... - group of 9 »
DE Culler, A Sah, KE Schauser, T von Eicken, J ... - Proceedings of the fourth international conference on ..., 1991 - portal.acm.org
... the initialization thread set up to re- **store** them from ... the structure to reflect different **scheduling** policies in ... 21] an I-structure fetch **instruction** issues a ...
Cited by 199 - Related Articles - Web Search - Library Search

Open research compiler (orc) 2.0 and tuning performance on itanium
R Ju, S Chan, TF Ngai, C Wu, Y Lu, J Zhang - 35th International Symposium on Microarchitecture, December, 2002 - cs.ualberta.ca
... gap from source language to machine **instructions** • Same IR ... eg MISTORE – indirect **store** of memory chunk • Field ID ... **Register** Variable Identification II ...
Cited by 4 - Related Articles - View as HTML - Web Search

Gooooooooogle ▶
Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

instrumentation instruction schedulir Search

Google Home - About Google - About Google Scholar